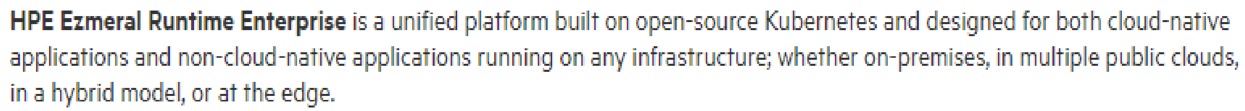
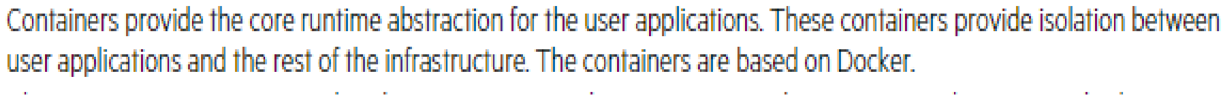
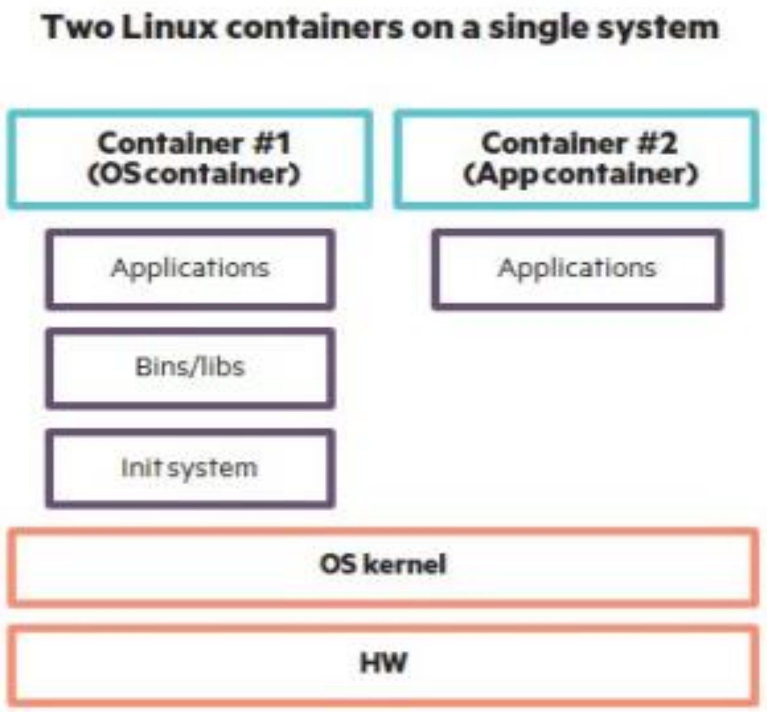
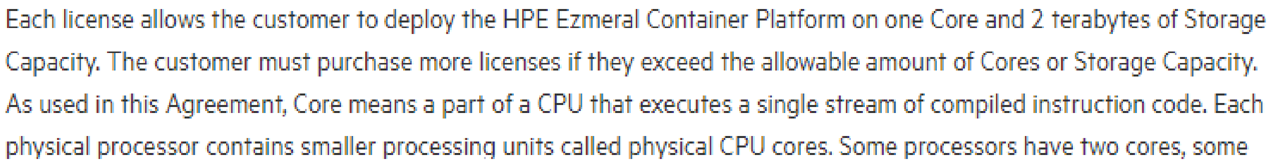
Exhibit 4

U.S. Patent No. 7,784,058 vs. HPE

Accused Instrumentalities: HPE's Ezmeral Runtime Enterprise, and all versions and variations thereof since the issuance of the asserted patent.

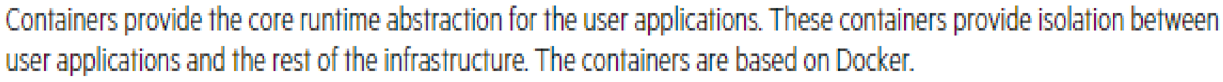
Claim 1

Claim 1	Accused Instrumentalities
[1pre] 1. A computing system for executing a plurality of software applications comprising:	To the extent the preamble is limiting, each Accused Instrumentality comprises or constitutes a computing system for executing a plurality of software applications as claimed. *See* claim limitations below. *See also, e.g.*:  **HPE Ezmeral Runtime Enterprise** is a unified platform built on open-source Kubernetes and designed for both cloud-native applications and non-cloud-native applications running on any infrastructure; whether on-premises, in multiple public clouds, in a hybrid model, or at the edge. https://support.hpe.com/hpesc/public/docDisplay?docId=a00ecp54hen_us&page=home/about-hpe-ezmeral-container-pl/Welcome.html  Containers provide the core runtime abstraction for the user applications. These containers provide isolation between user applications and the rest of the infrastructure. The containers are based on Docker. https://support.hpe.com/hpesc/public/docDisplay?docId=a00097165en_us&docLocale=en_US&page=GUID-6B6676DB-AF5F-4555-B6AB-D2C11A89F320.html

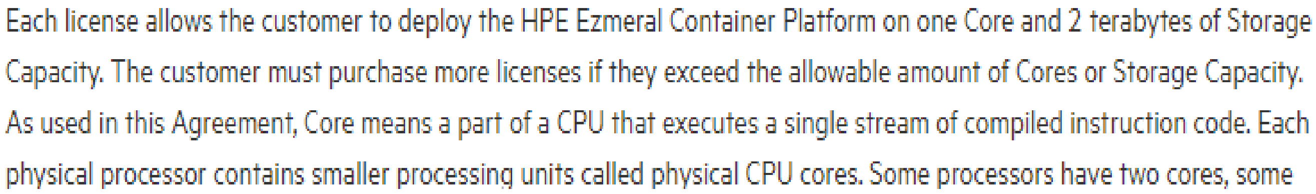
Claim 1	Accused Instrumentalities
	 Two Linux containers on a single system Container #1 (OS container) Container #2 (App container) Applications Applications Bins/libs Init system OS kernel HW https://h50146.www5.hpe.com/products/software/oe/linux/mainstream/support/whitepaper/pdfs/4AA6-2761ENW.pdf
[1a] a) a processor;	Each Accused Instrumentality comprises a processor. *See, e.g.*:  Each license allows the customer to deploy the HPE Ezmeral Container Platform on one Core and 2 terabytes of Storage Capacity. The customer must purchase more licenses if they exceed the allowable amount of Cores or Storage Capacity. As used in this Agreement, Core means a part of a CPU that executes a single stream of compiled instruction code. Each physical processor contains smaller processing units called physical CPU cores. Some processors have two cores, some https://support.hpe.com/hpesc/public/docDisplay?docId=a00ecp54hen_us&docLocale=en_US&page=home/about-hpe-ezmeral-container-pl/GEN_End_User_Software_Agreement.html

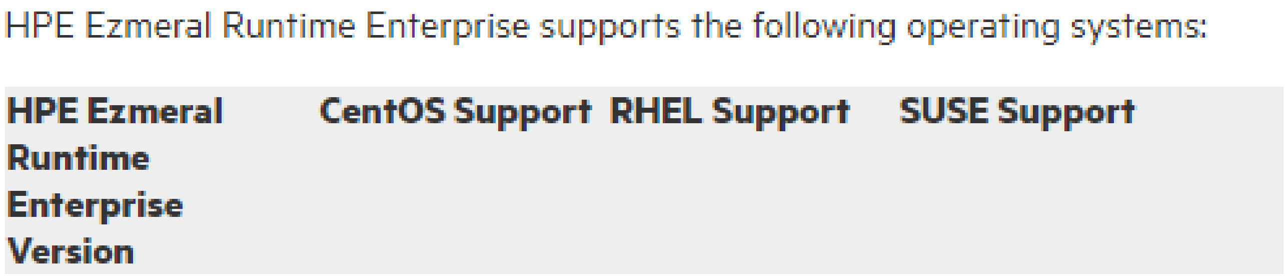
<table>
<tr><th>Claim 1</th><th>Accused Instrumentalities</th></tr>
<tr><td>[1b] b) an operating system having an operating system kernel having OS critical system elements (OSCSEs) for running in kernel mode using said processor; and,</td><td>Each Accused Instrumentality comprises an operating system having an operating system kernel having OS critical system elements (OSCSEs) for running in kernel mode using said processor.

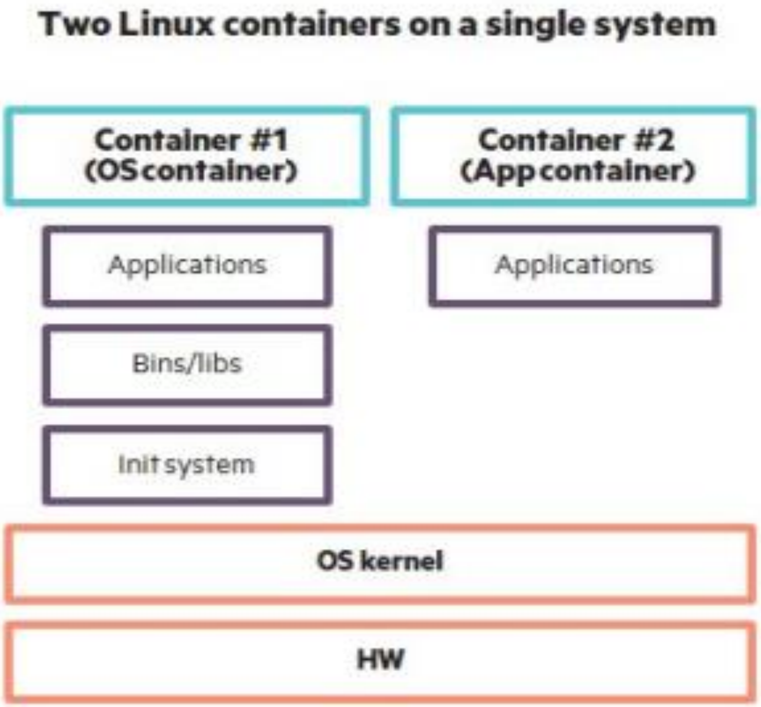
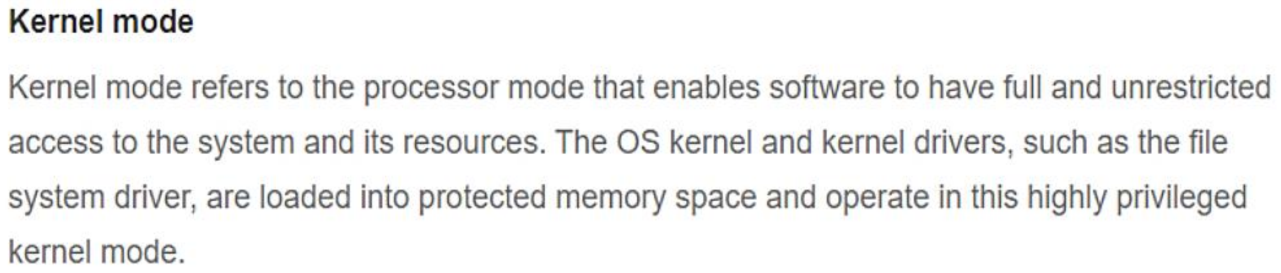
<em>See, e.g.</em>:

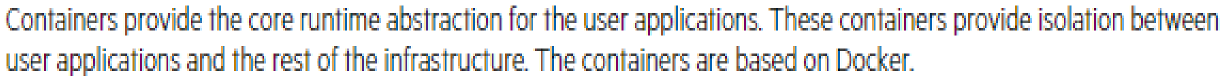
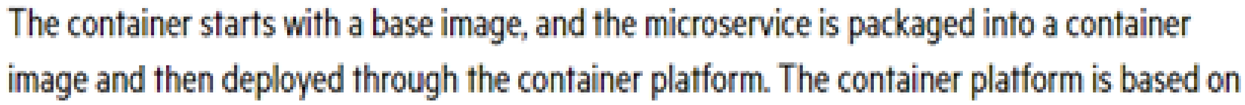
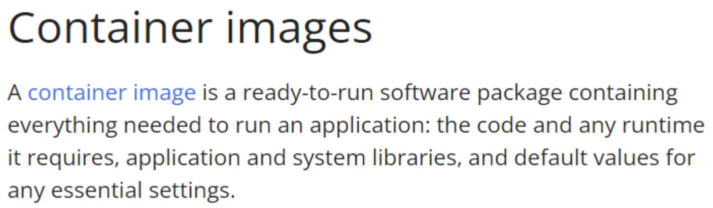
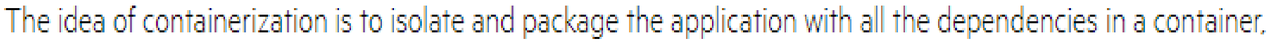

Containers provide the core runtime abstraction for the user applications. These containers provide isolation between user applications and the rest of the infrastructure. The containers are based on Docker.

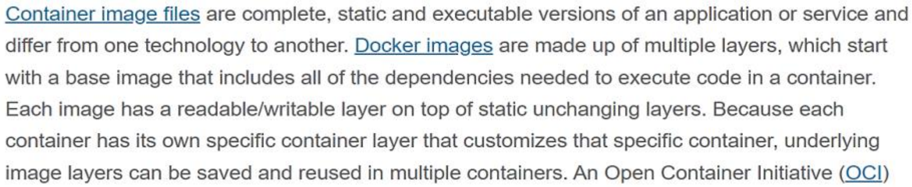
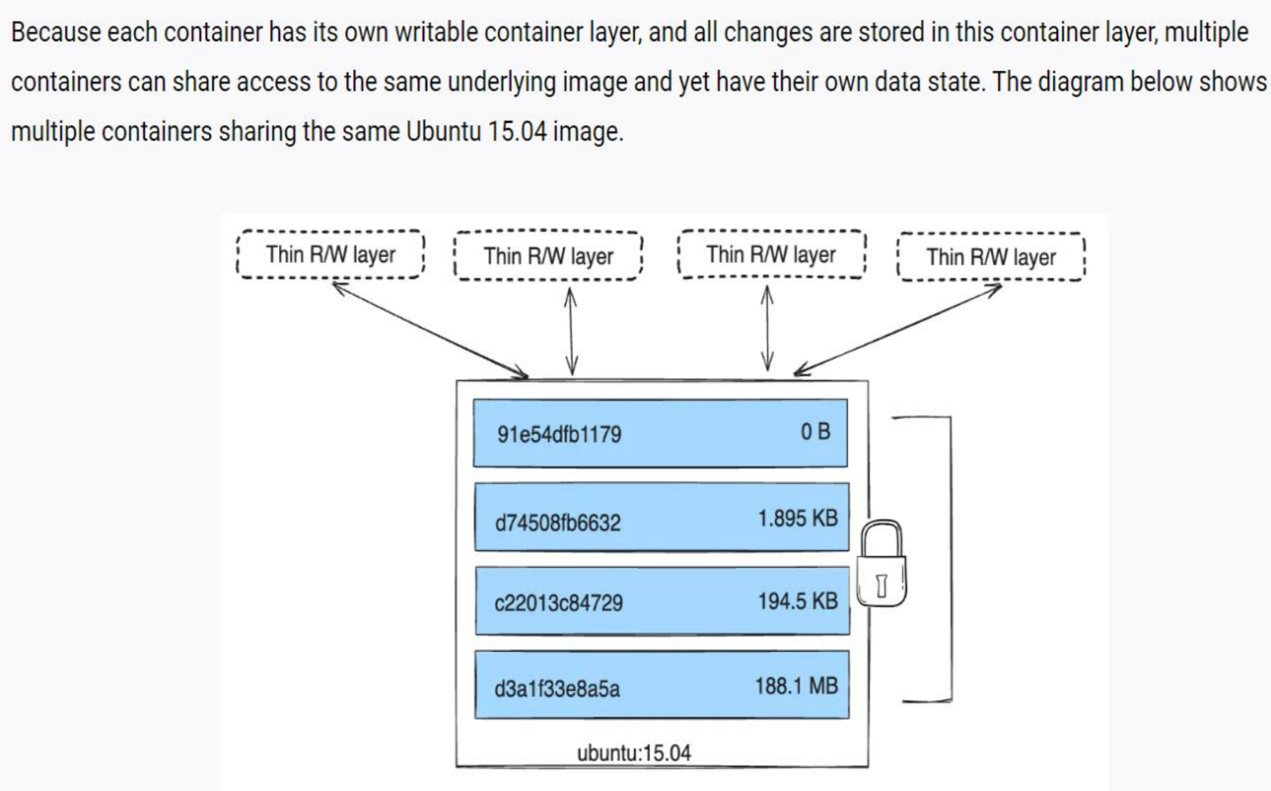
<a href="https://support.hpe.com/hpesc/public/docDisplay?docId=a00097165en_us&docLocale=en_US&page=GUID-6B6676DB-AF5F-4555-B6AB-D2C11A89F320.html">https://support.hpe.com/hpesc/public/docDisplay?docId=a00097165en_us&docLocale=en_US&page=GUID-6B6676DB-AF5F-4555-B6AB-D2C11A89F320.html</a>

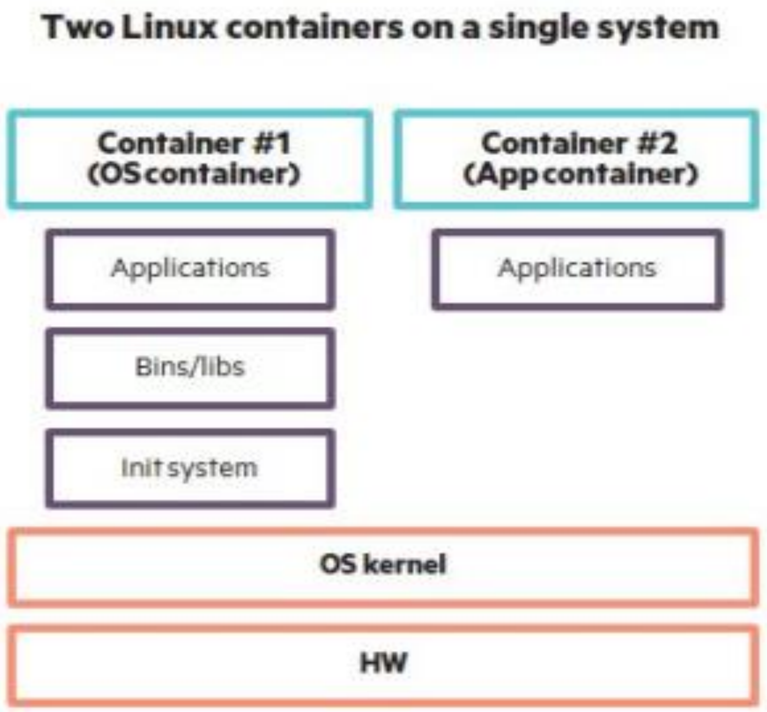

Each license allows the customer to deploy the HPE Ezmeral Container Platform on one Core and 2 terabytes of Storage Capacity. The customer must purchase more licenses if they exceed the allowable amount of Cores or Storage Capacity. As used in this Agreement, Core means a part of a CPU that executes a single stream of compiled instruction code. Each physical processor contains smaller processing units called physical CPU cores. Some processors have two cores, some
<a href="https://support.hpe.com/hpesc/public/docDisplay?docId=a00ecp54hen_us&docLocale=en_US&page=home/about-hpe-ezmeral-container-pl/GEN_End_User_Software_Agreement.html">https://support.hpe.com/hpesc/public/docDisplay?docId=a00ecp54hen_us&docLocale=en_US&page=home/about-hpe-ezmeral-container-pl/GEN_End_User_Software_Agreement.html</a>

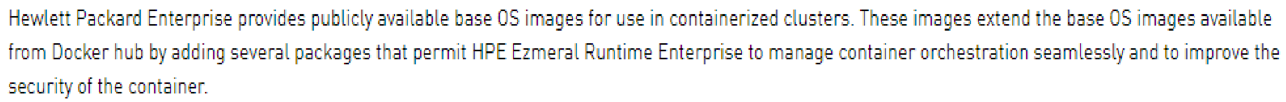
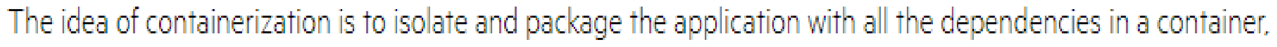
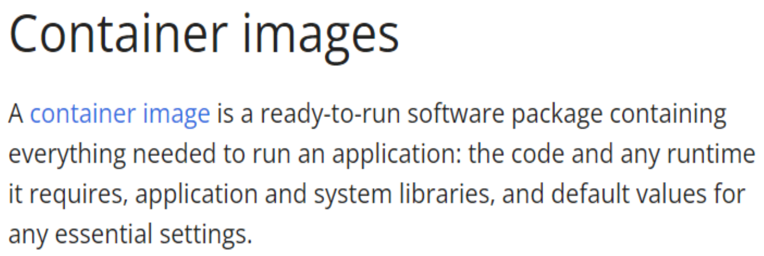
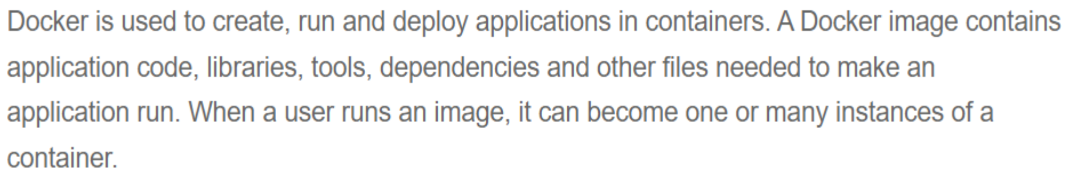

HPE Ezmeral Runtime Enterprise supports the following operating systems:
<table><tr><th>HPE Ezmeral Runtime Enterprise Version</th><th>CentOS Support</th><th>RHEL Support</th><th>SUSE Support</th></tr></table>
<a href="https://support.hpe.com/hpesc/public/docDisplay?docId=a00ecp54hen_us&page=home/about-hpe-ezmeral-container-pl/GEN_OS_Support.html">https://support.hpe.com/hpesc/public/docDisplay?docId=a00ecp54hen_us&page=home/about-hpe-ezmeral-container-pl/GEN_OS_Support.html</a></td></tr>
</table>

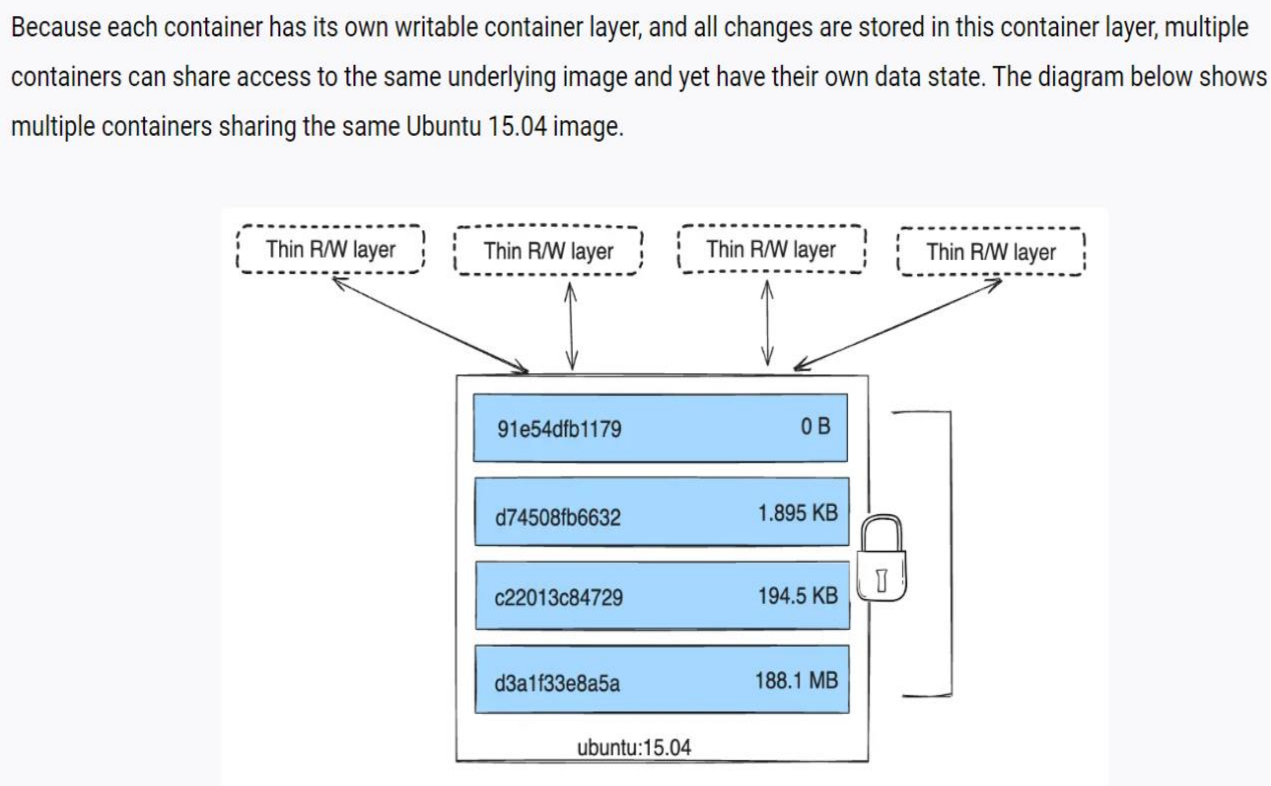
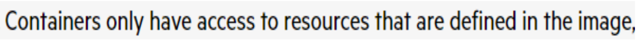
Claim 1	Accused Instrumentalities
	 Two Linux containers on a single system Container #1 (OS container) Container #2 (App container) Applications Applications Bins/libs Init system OS kernel HW https://h50146.www5.hpe.com/products/software/oe/linux/mainstream/support/whitepaper/pdfs/4AA6-2761ENW.pdf  **Kernel mode** Kernel mode refers to the processor mode that enables software to have full and unrestricted access to the system and its resources. The OS kernel and kernel drivers, such as the file system driver, are loaded into protected memory space and operate in this highly privileged kernel mode. https://www.techtarget.com/searchdatacenter/definition/kernel

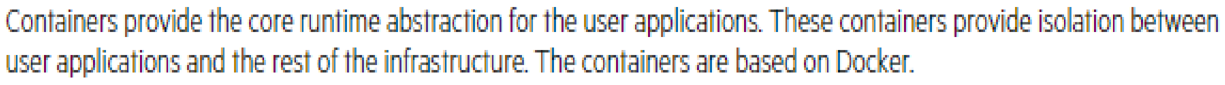
Claim 1	Accused Instrumentalities
[1c] c) a shared library having shared library critical system elements (SLCSEs) stored therein for use by the plurality of software applications in user mode and	Each Accused Instrumentality comprises a shared library having shared library critical system elements (SLCSEs) stored therein for use by the plurality of software applications in user mode. *See, e.g.*:  Containers provide the core runtime abstraction for the user applications. These containers provide isolation between user applications and the rest of the infrastructure. The containers are based on Docker. https://support.hpe.com/hpesc/public/docDisplay?docId=a00097165en_us&docLocale=en_US&page=GUID-6B6676DB-AF5F-4555-B6AB-D2C11A89F320.html  The container starts with a base image, and the microservice is packaged into a container image and then deployed through the container platform. The container platform is based on https://www.hpe.com/us/en/what-is/container-platform.html  Container images A container image is a ready-to-run software package containing everything needed to run an application: the code and any runtime it requires, application and system libraries, and default values for any essential settings. https://kubernetes.io/docs/concepts/containers/  The idea of containerization is to isolate and package the application with all the dependencies in a container. https://community.hpe.com/t5/hpe-blog-uk-ireland-middle-east/containerization-the-next-generation-of-virtualization/ba-p/7154442

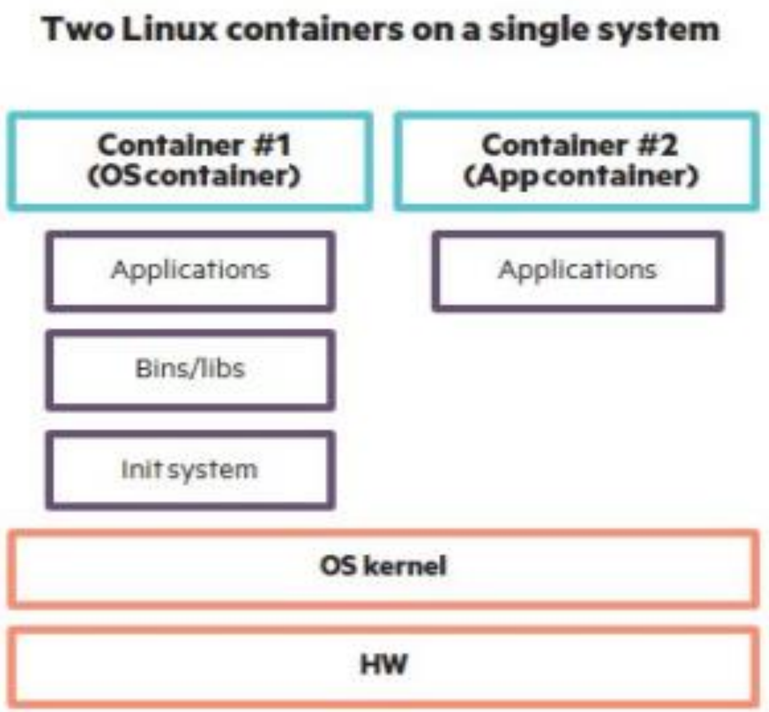
Claim 1	Accused Instrumentalities
	 Container image files are complete, static and executable versions of an application or service and differ from one technology to another. Docker images are made up of multiple layers, which start with a base image that includes all of the dependencies needed to execute code in a container. Each image has a readable/writable layer on top of static unchanging layers. Because each container has its own specific container layer that customizes that specific container, underlying image layers can be saved and reused in multiple containers. An Open Container Initiative (OCI) https://www.techtarget.com/searchitoperations/definition/container-containerization-or-container-based-virtualization  Because each container has its own writable container layer, and all changes are stored in this container layer, multiple containers can share access to the same underlying image and yet have their own data state. The diagram below shows multiple containers sharing the same Ubuntu 15.04 image. Thin R/W layer Thin R/W layer Thin R/W layer Thin R/W layer 91e54dfb1179 0 B d74508fb6632 1.895 KB c22013c84729 194.5 KB d3a1f33e8a5a 188.1 MB ubuntu:15.04 https://docs.docker.com/storage/storagedriver/

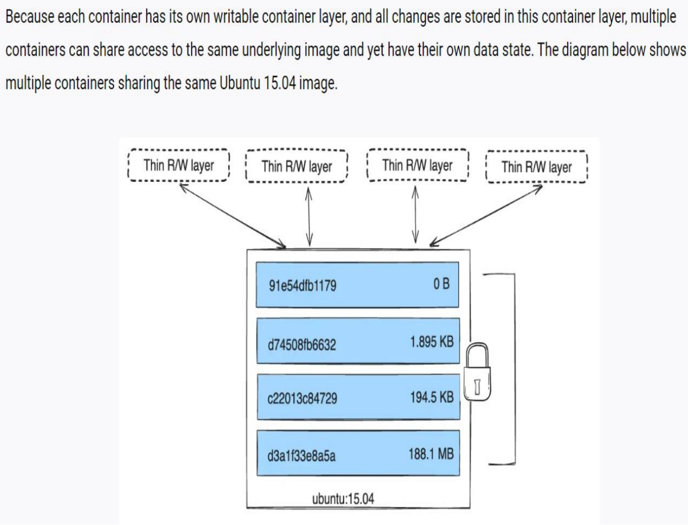
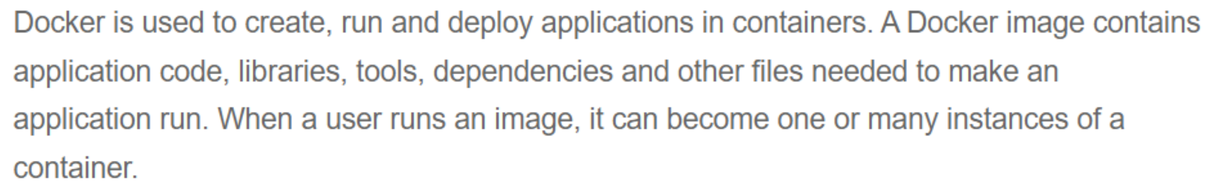
Claim 1	Accused Instrumentalities
	 Two Linux containers on a single system Container #1 (OS container) Container #2 (App container) Applications Applications Bins/libs Init system OS kernel HW https://h50146.www5.hpe.com/products/software/oe/linux/mainstream/support/whitepaper/pdfs/4AA6-2761ENW.pdf
[1d] i) wherein some of the SLCSEs stored in the shared library are functional replicas of OSCSEs and are accessible to some of the plurality of software applications and when one of the SLCSEs is accessed by one or more of the plurality of software applications it forms a part of the one or more of the plurality of software applications,	In each Accused Instrumentality, some of the SLCSEs stored in the shared library are functional replicas of OSCSEs and are accessible to some of the plurality of software applications and when one of the SLCSEs is accessed by one or more of the plurality of software applications it forms a part of the one or more of the plurality of software applications. For example, a Docker base image serves as a self-contained unit that encompasses all the necessary components for an application to run, including the application code, runtime environment, system tools, and dependencies (i.e., SLCSEs). The images are based on existing Linux distributions, such as Debian and Ubuntu, including essential system elements (i.e., functional replicas of OSCSEs). Each container image is based on a specific base image, which contains the application code, and dependencies, including system libraries or shared library critical system elements (SLCSEs). When the container runs the image, it creates a runtime instance of that container image.

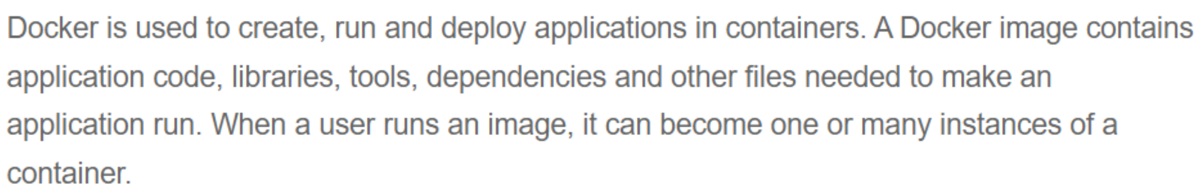
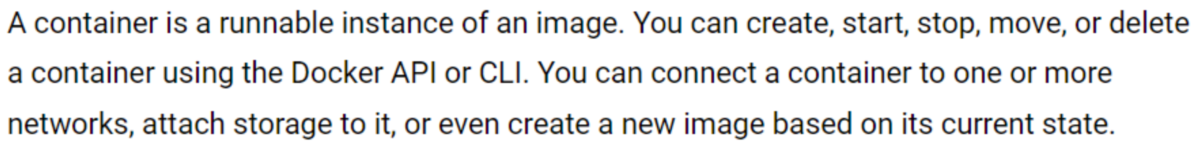
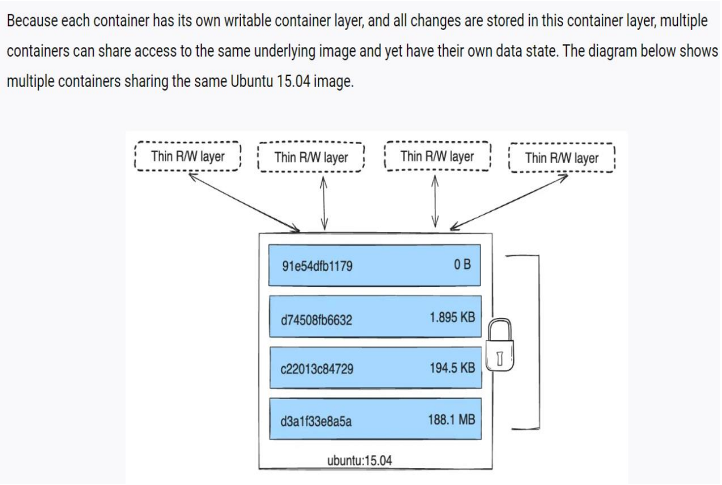
Claim 1	Accused Instrumentalities
	See, e.g.:  Hewlett Packard Enterprise provides publicly available base OS images for use in containerized clusters. These images extend the base OS images available from Docker hub by adding several packages that permit HPE Ezmeral Runtime Enterprise to manage container orchestration seamlessly and to improve the security of the container. https://docs.ezmeral.hpe.com/runtime-enterprise/55/app-workbench-5-1/custom-base-images/AWB51_About_Custom_Base_Images.html  The idea of containerization is to isolate and package the application with all the dependencies in a container. https://community.hpe.com/t5/hpe-blog-uk-ireland-middle-east/containerization-the-next-generation-of-virtualization/ba-p/7154442  Container images A container image is a ready-to-run software package containing everything needed to run an application: the code and any runtime it requires, application and system libraries, and default values for any essential settings. https://kubernetes.io/docs/concepts/containers/  Docker is used to create, run and deploy applications in containers. A Docker image contains application code, libraries, tools, dependencies and other files needed to make an application run. When a user runs an image, it can become one or many instances of a container. ttps://www.techtarget.com/searchitoperations/definition/Docker-image

Claim 1	Accused Instrumentalities
	 Because each container has its own writable container layer, and all changes are stored in this container layer, multiple containers can share access to the same underlying image and yet have their own data state. The diagram below shows multiple containers sharing the same Ubuntu 15.04 image. Thin R/W layer Thin R/W layer Thin R/W layer Thin R/W layer 91e54dfb1179 0 B d74508fb6632 1.895 KB c22013c84729 194.5 KB d3a1f33e8a5a 188.1 MB ubuntu:15.04 https://docs.docker.com/storage/storagedriver/  Containers only have access to resources that are defined in the image, https://www.hpe.com/us/en/what-is/docker.html
[1e] ii) wherein an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library is run in a context of said at least first of the plurality of software applications without being shared with other of the plurality of software applications and where at least a second of the plurality of software applications	In each Accused Instrumentality, an instance of a SLCSE provided to at least a first of the plurality of software applications from the shared library is run in a context of said at least first of the plurality of software applications without being shared with other of the plurality of software applications and where at least a second of the plurality of software applications running under the operating system have use of a unique instance of a corresponding critical system element for performing same function. When a Docker image is used to create a container, it creates a separate and isolated instance of a runtime environment which is independent of other containers running on the same host. Each

Claim 1	Accused Instrumentalities
running under the operating system have use of a unique instance of a corresponding critical system element for performing same function, and	container has its own instance of base images and its own data. The containers run in isolation, ensuring that the SLCSEs stored in the shared library are accessible to the software applications running in their respective containers. The docker image includes essential system files, libraries, and dependencies required to run the software application within the container. The Docker containers can share common dependencies and components using layered images. This means that multiple containers utilize the same base image to create an instance. When an instance of SLCSE is provided from the base image (i.e., from the shared library) to an individual container including application software, it operates in isolation and runs its own instance of the software application without sharing resources or critical system elements with other containers. This ensures that each container has its own isolated context. Docker containers can share common dependencies and components using layered images. This means that multiple containers can utilize the same base image. Therefore, each container, containing the application software running under the operating system, utilizes a unique instance of the corresponding critical system element to execute the respective application software for performing a same or a different function. *See, e.g.*:  Containers provide the core runtime abstraction for the user applications. These containers provide isolation between user applications and the rest of the infrastructure. The containers are based on Docker. https://support.hpe.com/hpesc/public/docDisplay?docId=a00097165en_us&docLocale=en_US&page=GUID-6B6676DB-AF5F-4555-B6AB-D2C11A89F320.html

Claim 1	Accused Instrumentalities
	 Two Linux containers on a single system Container #1 (OS container) Container #2 (App container) Applications Applications Bins/libs Init system OS kernel HW https://h50146.www5.hpe.com/products/software/oe/linux/mainstream/support/whitepaper/pdfs/4AA6-2761ENW.pdf

Claim 1	Accused Instrumentalities
	 Because each container has its own writable container layer, and all changes are stored in this container layer, multiple containers can share access to the same underlying image and yet have their own data state. The diagram below shows multiple containers sharing the same Ubuntu 15.04 image. Thin R/W layer Thin R/W layer Thin R/W layer Thin R/W layer 91e54dfb1179 0 B d74508fb6632 1.895 KB c22013c84729 194.5 KB d3a1f33e8a5a 188.1 MB ubuntu:15.04 https://docs.docker.com/storage/storagedriver/  Docker is used to create, run and deploy applications in containers. A Docker image contains application code, libraries, tools, dependencies and other files needed to make an application run. When a user runs an image, it can become one or many instances of a container. https://www.techtarget.com/searchitoperations/definition/Docker-image
[1f] iii) wherein a SLCSE related to a predetermined function is provided to the first of the plurality of software applications for running a first instance of the SLCSE, and wherein a SLCSE for performing a same function is provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously.	In each Accused Instrumentality, a SLCSE related to a predetermined function is provided to the first of the plurality of software applications for running a first instance of the SLCSE, and wherein a SLCSE for performing a same function is provided to the second of the plurality of software applications for running a second instance of the SLCSE simultaneously. For example, In Docker, each container operates independently, and a Docker base image includes essential system files, libraries, and dependencies (i.e., SLCSEs) required to run the software application within the container. Based on information and belief, each element, such as system files, libraries, and dependencies (i.e., SLCSE) is associated with an execution of a predetermined function related to the application. When a Docker image is used to create a container in ECS, an instance of

Claim 1	Accused Instrumentalities
	the SLCSE is provided to a software application. Therefore, different instances of the SLCSE are provided to different applications for performing either a same or a different function, simultaneously. *See, e.g.*:  Docker is used to create, run and deploy applications in containers. A Docker image contains application code, libraries, tools, dependencies and other files needed to make an application run. When a user runs an image, it can become one or many instances of a container. https://www.techtarget.com/searchitoperations/definition/Docker-image, Last accessed on June 14, 2023  A container is a runnable instance of an image. You can create, start, stop, move, or delete a container using the Docker API or CLI. You can connect a container to one or more networks, attach storage to it, or even create a new image based on its current state. https://docs.docker.com/get-started/overview/  Because each container has its own writable container layer, and all changes are stored in this container layer, multiple containers can share access to the same underlying image and yet have their own data state. The diagram below shows multiple containers sharing the same Ubuntu 15.04 image. Thin R/W layer Thin R/W layer Thin R/W layer Thin R/W layer 91e54dfb1179 0 B d74508fb6632 1.895 KB c22013c84729 194.5 KB d3a1f33e8a5a 188.1 MB ubuntu:15.04 https://docs.docker.com/storage/storagedriver/